MATLAB: Plotting Data & Formatting

Written By Abi Chiaokhiao

This lesson assumes you have know or have read the basics.

Files: W2_StartFile.m

Disclaimer: Many examples taken from mathworks.com

Table of Contents

Line Plots

Editing Plots

Properties

The ln keyword shows you the properties of the given plot

- To access any, use the dot operator: `objName.Color` ⇒ `0 0 1`
  - `ln` can also be used in place of `objName`

To add min & max:

1. `indexOfMax = find(y == max(y));`
2. `indexOfMin = find(y == min(y));`
3. Add 'MarkerIndices' & `[indexOfMax indexOfMin otherPts]`
   a. A good way to do the `otherPts` is using colons ⇒ `start:step:end`
      i. So if you put `1:2:10`, you're saying start at the first point, show every 2 points, until you've gotten to index 10 points (in MATLAB, it starts at 1, not 0 like in other coding situations)

Uncertainty

Adding a Shaded Area of Uncertainty

- Note: make sure the x & y values align in their respective arrays

1. Make x bounds: `xconf = [xVals xValsBackwards];`
   a. `xValsBackwards is done by doing xVals(end:-1:1)`
2. Make y bounds: `yconf = [y+confVal yValsBackwards-confVal];`
   a. `yValsBackwards is done by doing yVals(end:-1:1)`

- These bounds are arrays that are zipped together to make x,y coordinates
- The reason you have to have one part of the arrays be the values going backwards is so that when each point is made and connected to the next, it is drawn correctly

3. Use `fill` function: `fill(xCoords, yCoords, Color)`
   a. Note: if one of the properties is FaceAlpha, you can change the transparency

Adding Error Bars (more here)

1. Initialize your x and y values
2. Initialize error as an array where each element correlates to a point on the graph
   a. For error consistent for every point: `error = errorVal*ones(size(y))`
3. Use `errorbar(x,y,error)` function
   a. To make them horizontal, add `'horizontal'` as a fourth argument to `errorbar`, if you want both, type `'both'` instead
   b. If you don't want a line, add a marker shape spec (ex: `'o'` for circles)

Line Plots (con.)

Axes Formatting (more on axes here)

Changing x-axis Formatting

- To know your x-axis' limits, use `xlim`

Having Multiple y-axes

- Type `yyaxis left` or `yyaxis right`. This will activate having two y axes.
- Depending on the last axis to be denoted (left or right), that is where the following will be executed.
  - Labeling is done this way
  - If you want to clear one axis, you write `cla` when that axis is active in your code

Bar Data

Histograms

A type of bar plot that puts data into "bins".
Using `histogram()`, a vector of data can be sorted

Example
```
h=histogram(x, nbins');
h.FaceColor = 'g';
```

- You can set properties using the object name (in the above case "h")

You can also plot categorical data:

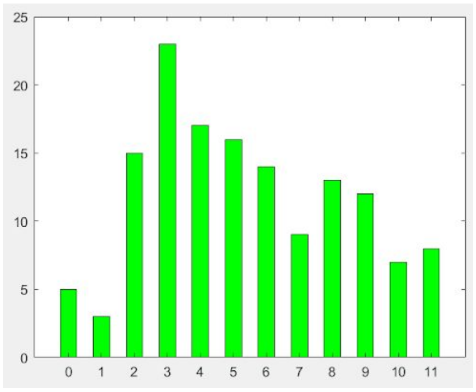
Example
```
A = [0 0 1 1 1 0 0 0 0 NaN NaN 1 0 0 0…
     1 0 1 0 1 0 0 0 1 1 1 1];
C = categorical(A,[1 0 NaN],{'yes','no','undecided'})
```

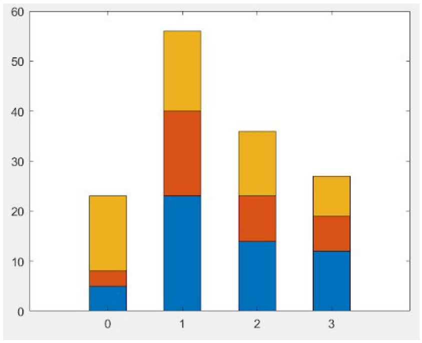
- You can also do other things like create edges to capture outliers, plot multiple on one graph, or plot a line over it. More here

Bar graph

Takes x & y data and plots them in bars
A couple attributes that can be changed are bar width & facecolor

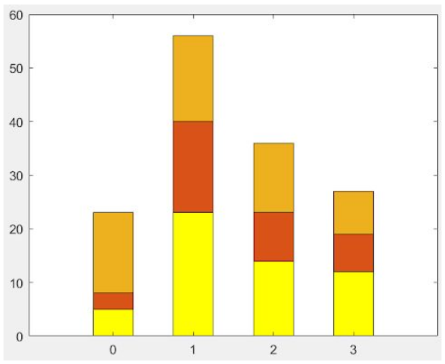
Example
```
barX = 0:11;
barY = [5 3 15 23 17…
16 14 9 13 12 7 8];
b = bar(barX,barY, 0.5,
'FaceColor', 'g')
```





The bars can also be grouped together & stacked

Example
```
barX = 0:3;
barY = [5 3 15; 23 17 16; 14 9 13; 12 7
8];
b = bar(barX,barY, 0.5, 'stacked',
'FaceColor', 'flat')
```



To change up colors, the RGB can be set for a stack layer

Example `b(1).CData = [1 1 0] % Yellow`

More colors here

Note: `categorical()` can be used on bar graphs, find that and other info here. There are also 3D Bar graphs, a color by height example is here

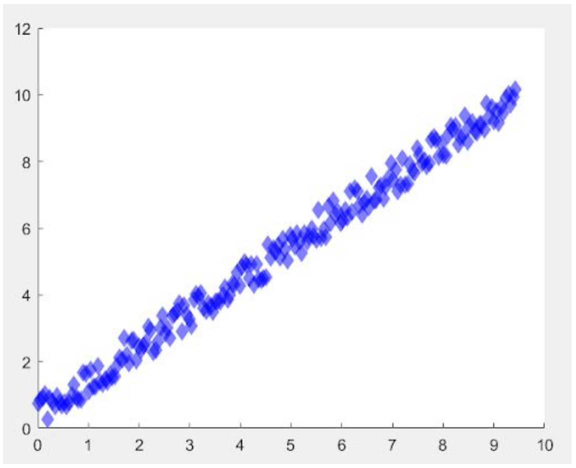
Other Data Plots

Scatter Plots

Points of x & y coordinates are plot with no line connecting them.
Starts off with the `scatter(xData, yData)` function, other attributes such as size, color, and marker shape can be added.

Example

```
x = linspace(0,3*pi,200);
y = x + rand(1,200);
sz = 75*ones(1,length(sPX));
s =
scatter(x,y,sz,'filled','d',
'MarkerFaceColor','b')
s.MarkerFaceAlpha = 0.5;
```



More documentation here.

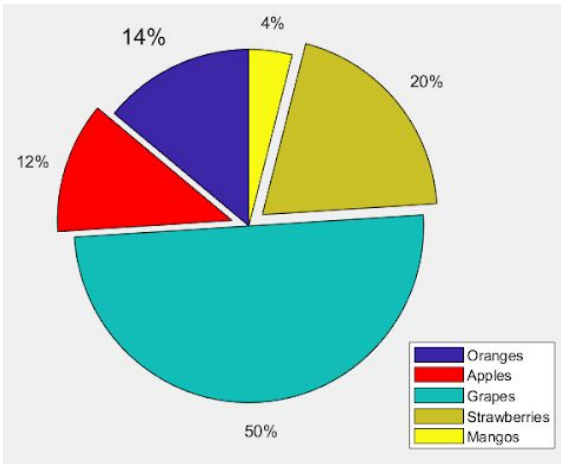
- For more customization, the size and the color can be varied over the points by assigning integers (one good way is using `linspace()`)

Pie Graphs

Using one vector of data, a pie graph is made with each vector element as a slice.
The base is `pie(pData)`, other things such as exploding slices off & labels can be changed in the function call.

Example

```
pData = [7 6 25 10 2];
labels = {'Oranges', 'Apples',
'Grapes', 'Strawberries',
'Mangos'};
p = pie(pData, [0 1 0 1 0]);
legend(labels);
p(3).FaceColor = 'r';
p(2).FontSize = 14;
```



The graphics vector interchanges between slice & text, so to access properties of each specific elements need to be called (`p(3)` & `p(2)` were examples above).
Some more pie graph documentation.
Another cool plot is a quiver or vector plot if you would ever like to play around with those.

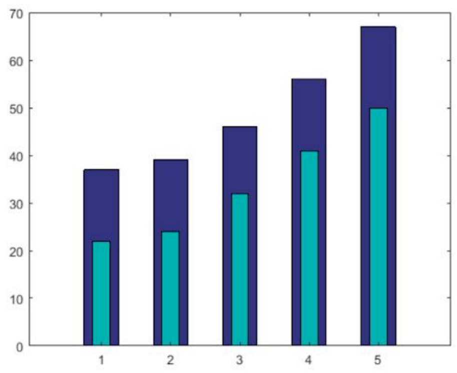
Overlaid Graphs

Bar Graph Overlay

If you want to show two bar datas in the same plot to compare them, you can overlay them.

Example

```
x = [1 2 3 4 5];
temp_high = [37 39 46 56 67];
w1 = 0.5;
bar(x, temp_high, w1,
'FaceColor', [0.2 0.2 0.5])
temp_low = [22 24 32 41 50];
w2 = .25;
hold on
bar(x,temp_low,w2,'FaceColor',
[0 0.7 0.7])
hold off
```



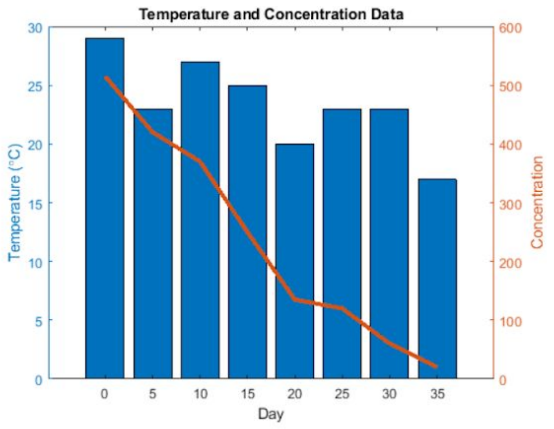
The function `categorical()` can also be used, more here.

Line + Bar Graphs (2 y-axes)

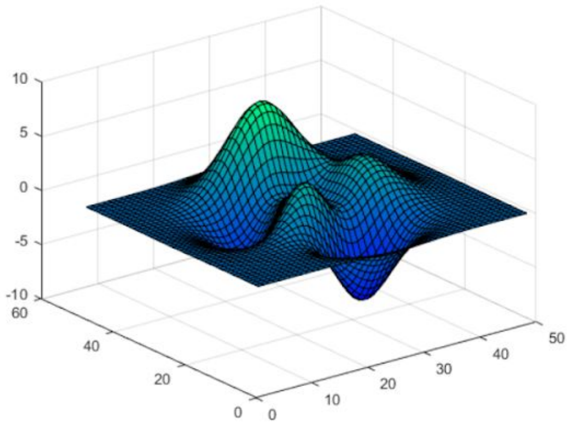
Can show a linear trend alongside bar data

Example

```
days = 0:5:35;
conc = [515 420 370 250 135
120 60 20];
temp = [29 23 27 25 20 23 23
17];
yyaxis left
b = bar(days,temp);
yyaxis right
p = plot(days,conc, ‘LineWidth’, 3);
```



Another type is area graphs if you want to be able to make those. There’s also the aforementioned quiver/vector plots that can be overlaid onto contours.



More on Color Formatting

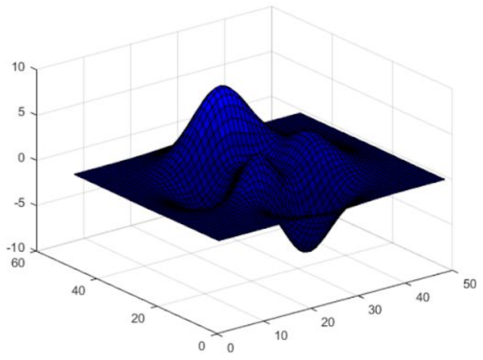
Colormaps

Colormaps are already in MATLAB to give a certain range of color.

Example

```
surf(peaks)
colormap winter
```

To make your own, have a 3 column rgb matrix and use `colormap()`.



Example

```
map = [0 0 0.3
       0 0 0.4
       0 0 0.5
       0 0 0.6
       0 0 0.8
       0 0 1.0];
surf(peaks)
colormap(map)
```

More information and specific colormaps here.

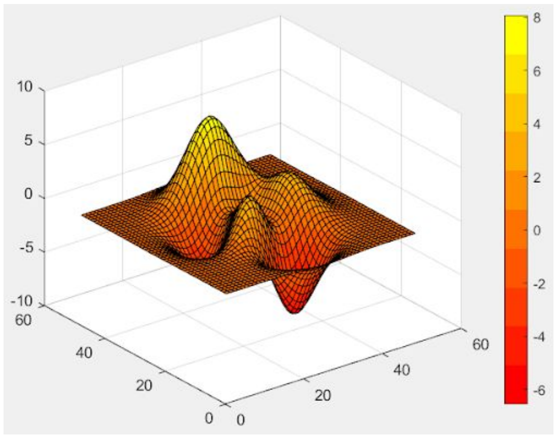
Colorbar

To get a legend, just type colorbar in a line of its own.

Example

```
surf(peaks)
colormap(autumn(10))
colorbar
```

Note: the number 10 is saying how many intervals using the colormap autumn

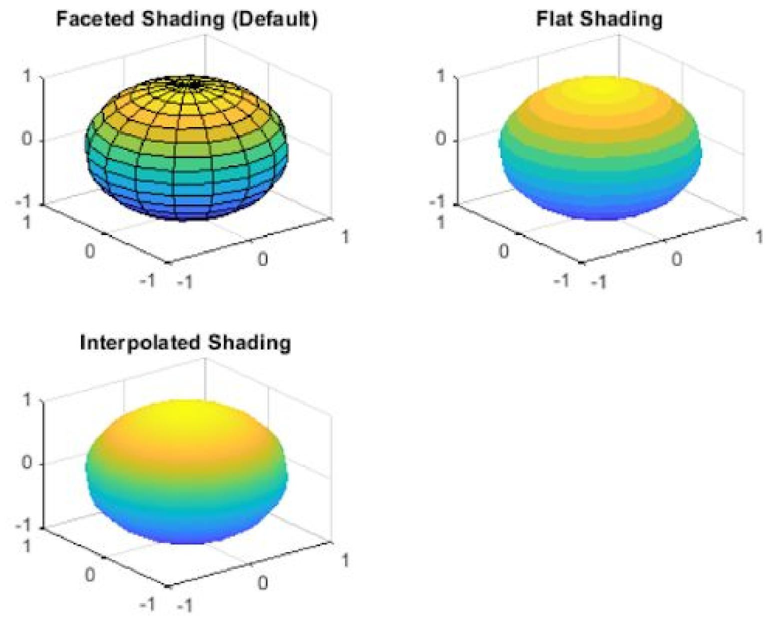


More on Color Formatting (con.)

Shading

To give different shading effects, the keyword `shading` is used.

- `flat`: line segments and their faces have similar colors
- `faceted`: each line is a black line, making a mesh
- `interp`: color is interpolated to make a smooth look



Functions (pg 1/3)

Input	Output	Description	Section
`find(`*`value`*`)`	Number	The index of `value` is returned	Line Plots - Editing Plots
`fill(`*`xCoords, yCoords, Color`*`)`	2-D Colored area	Using an array of x components (`xCoords`), an array of y components (`yCoords`), and a linespec (`Color`), an area shaded with the given `Color`	Line Plots - Uncertainty
`ones(`*`size`*`)`	An integer array	An array, all of value 1, is made of size `size`	
`size(`*`val`*`)`	Number	Returns the size of `val`	
`errorbar(`*`x,y, error`*`)`	Error bars	Using an array of x components (`x`), an array of y components (`y`), & an array of error values for each coordinate (`error`), error bars are made on each point. *Optional: to make them horizontal, have a 4th argument `'horizontal'`; for both horizontal & vertical, this would change to `'both'`. Linespecs can also be added as inputs	
`xlim`	A string	Prints the limits and units (if applicable) of the x-axis	Line Plots - Axes Formatting
`datetime(`*`date`*`)`	Number-String-Number	A date is created using `date`. Can be done like (2020, 08, 19) or other things like ('today'). More here	
`calweeks(`*`startWk: endWk`*`)`	An array	An array is created where each element is written in the form of "#w" where the first # is `startWk`, then it goes up by 1 each element until the number is `endWk`	
`xtickformat(`*`format`*`)`	Format change	The x-axis values are formatted in such a way that `format` dictates	

Functions (pg 2/3)

Input	Output	Description	Section
`seconds(val)`	A string	A string which represents seconds is made in the format "`val sec`"	Line Plots - Axes Formatting
`minutes(val)`	A string	A string which represents seconds is made in the format "`val min`"	
`cla`	Nothing	The active code is cleared from the returned data. You can also write the keyword `reset` after to reset it	
`yyaxis axis`	Y axes	Keyword to determine whether you're working on the left or right `axis`	
`histogram(data, …)`	Plot	Takes `data` and organizes them in a plot of bins, characteristics can be set in function or later using the object name	Bar Data
`categorical( data, [#s],{ 'newString1' 'newString2' …})`	Remapped vector	Takes `data` and replaces the corresponding `#s` with the respective `newStrings`	
`bar(xData,yData , ...)`	Bar graph	Each element of `xData` is made into a bar of corresponding `yData` height. Attributes such as `FaceColor` can be added in the function call.	
`scatter(xData, yData, ...)`	Scatter plot	Each element of `xData` is paired to make a point using corresponding `yData`. Attributes such as `MarkerFaceColor` can be added in the function call.	Other Data Plots
`pie(pData, ...)`	Pie graph	Using `pData`, a pie graph is made with each vector element as a slice.	
`colormap map`	Plot coloring	Keyword to use predetermined color schemes `map` on a plot.	More on Color Formatting

Functions (pg 3/3)

Input	Output	Description	Section
`colormap(map)`	Plot coloring	Function to use a predetermined color scheme in place of `map`, or a customized one.	More on Color Formatting
`colorbar`	A colorbar	A color legend bar is added to an existing plot.	
`shading type`	Plot shading	Keyword to change the shading of a plot according to `type`.	

Additional Resources

In Document

Resource	Description	Section
Line Specs	Line specifications that can be used when plotting lines	Line Plots
Line Properties	Line properties that can be used when plotting lines	
Color Specs	Specifications for when changing/using color	Any
rgbColorCode.com	An RGB converter (use the ones with values 0-1)	
Plotting imaginary & complex data		
More MATLAB plot types		